

9 BAB IX LAYOUTING

9.1 IDENTITAS

Kajian

Komponen Swing Java non-Visual Editor; Database & Swing

Topik

1. Layout—Grid, GridBag dan Border

Referensi

1. <http://docs.oracle.com/javase/tutorial/uiswing/layout/index>

Kompetensi Utama

1. Mahasiswa memahami konsep layout pada pemrograman swing
2. Mahasiswa mampu membuat halaman sederhana menggunakan komponen swing dan mengaturnya menggunakan layout tertentu (grid, gridbag atau border) tanpa dibantu tool gui builder

Lama Kegiatan Praktikum

1. Kegiatan Mandiri : 1 x 100 menit

Parameter Penilaian

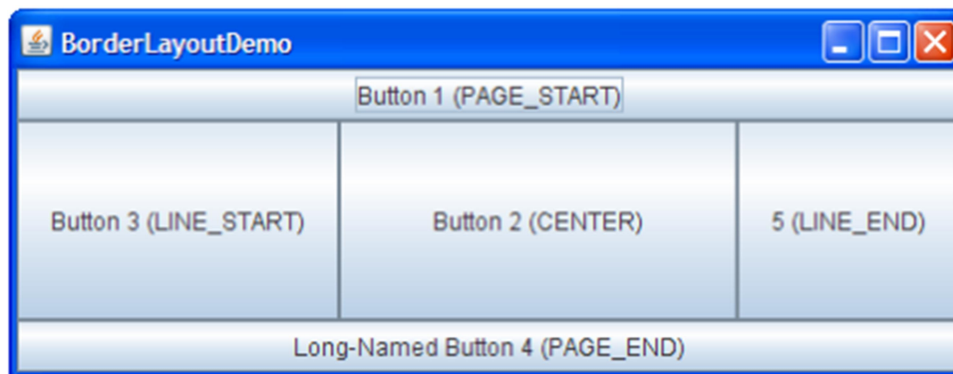
1. Tugas Pendahuluan
2. Jurnal Mandiri

9.2 BORDER LAYOUT

Layout yang memungkinkan komponen hanya dapat diletakkan di 5 area saja:

- PAGE_START
- PAGE_END
- LINE_START
- LINE_END
- CENTER

Ilustrasi:



Contoh Penggunaan:

```
/**
 *
 * edit By Eja
 */
import javax.swing.*.*;
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.Dimension;

public class BorderLayoutDemo {

    public static void addComponentsToPane(Container pane) {

        JButton button = new JButton("Button 1 (PAGE_START)");
        pane.add(button, BorderLayout.PAGE_START);

        //Make the center component big, since that's the
        //typical usage of BorderLayout.
        button = new JButton("Button 2 (CENTER)");
        button.setPreferredSize(new Dimension(200, 100));
        pane.add(button, BorderLayout.CENTER);
```

```

button = new JButton("Button 3 (LINE_START)");
pane.add(button, BorderLayout.LINE_START);

button = new JButton("Long-Named Button 4 (PAGE_END)");
pane.add(button, BorderLayout.PAGE_END);

button = new JButton("5 (LINE_END)");
pane.add(button, BorderLayout.LINE_END);

}
private static void createAndShowGUI() {

    //membuat frame
    JFrame frame = new JFrame("BorderLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //tambahkan pane
    addComponentsToPane(frame.getContentPane());
    //tampilkan frame
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {

    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

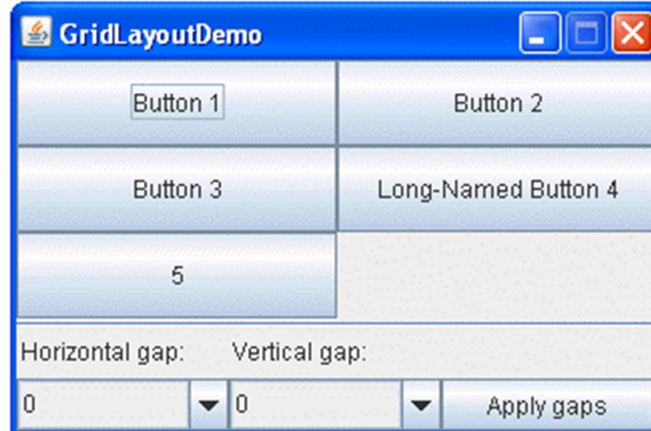
```

Perhatikan pada contoh di atas bahwa border layout tidak dibentuk objeknya sama sekali. Hal ini karena pane yang digunakan (JFrame) menggunakan default layout BorderLayout, jadi komponen bisa langsung ditambahkan dengan pengaturan manajemen layout "BorderLayout".

9.3 GRID LAYOUT

Grid layout menempatkan objek komponen berdasarkan grid cell. Dengan setiap cell memiliki ukuran yang sama.

Ilustrasi:



Terdapat beberapa method yang digunakan pada layout ini:

Method	Fungsi
setRows(3)	Menge-set jumlah maksimal baris dari peletakan komponen.
setColumns(4)	Menge-set jumlah maksimal kolom dari peletakan komponen. Orientasi setRows lebih besar dibandingkan setColumns.
setHgap(30)	Menge-set jarak antar komponen secara horizontal
setVgap(30)	Menge-set jarak antar komponen secara vertikal

Jumlah baris dan kolom maksimum pada layout juga dapat di-set via parameter konstruktor. Parameter pertama adalah nilai baris maksimal, parameter kedua merupakan pengaturan nilai kolom maksimal.

Contoh Penggunaan:

```
/**
 *
 * Edit By Eja
 */
import java.awt.*;
import javax.swing.*;

public class GridLayoutDemo extends JFrame {
```

```
final static int maxGap = 20;
//deklarasi layout dengan baris tidak ditentukan dan kolom maksimal 2
GridLayout experimentLayout = new GridLayout(0, 2);

public GridLayoutDemo(String name) {
    super(name);
}

public void addComponentsToPane(final Container pane) {
    final JPanel compsToExperiment = new JPanel();
    compsToExperiment.setLayout(experimentLayout);

    //tambahkan tombol ke panel
    compsToExperiment.add(new JButton("Button 1"));
    compsToExperiment.add(new JButton("Button 2"));
    compsToExperiment.add(new JButton("Button 3"));
    compsToExperiment.add(new JButton("Long-Named Button 4"));
    compsToExperiment.add(new JButton("5"));
    //atur jarak vertical dan horizontal-nya
    experimentLayout.setHgap(20);
    experimentLayout.setVgap(5);

    //mengatur letak panel pada objek JFrame
    pane.add(compsToExperiment, BorderLayout.NORTH);
}

private static void createAndShowGUI() {
    //membuat objek frame
    GridLayoutDemo frame = new GridLayoutDemo("GridLayoutDemo");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //memanggil method buatan addComponentsToPane()
    frame.addComponentsToPane(frame.getContentPane());
    //menampilkan window
    frame.pack();
    frame.setVisible(true);
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {

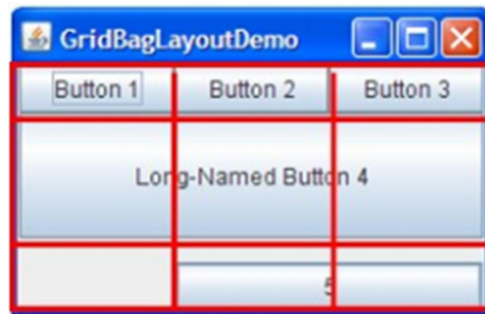
        public void run() {
            createAndShowGUI();
        }
    });
}
```

}

9.4 GRIDBAG LAYOUT

Merupakan layout yang paling sering digunakan programmer java untuk pengaturan peletakan komponen objek swing karena fleksibilitas yang ditawarkan. Sama seperti grid layout, membagi peletakan dalam grid cells. Hanya saja ukuran dari setiap komponen dapat berbeda. Contohnya, saat melakukan peletakan komponen, komponen tersebut dapat memakai 2 grid secara horizontal dan 3 grid vertical.

Ilustrasi:



Contoh Penggunaan:

```
import java.awt.*;
import javax.swing.JButton;
import javax.swing.JFrame;

public class GridBagLayoutDemo {
    final static boolean shouldFill = true;
    final static boolean shouldWeightX = true;
    final static boolean RIGHT_TO_LEFT = false;

    public static void addComponentsToPane(Container pane) {
        if (RIGHT_TO_LEFT) {
            pane.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
        }

        JButton button;
        pane.setLayout(new GridBagLayout());
        GridBagConstraints c = new GridBagConstraints();
        if (shouldFill) {
            //natural height, maximum width
            c.fill = GridBagConstraints.HORIZONTAL;
        }
    }
}
```

```
        button = new JButton("Button 1");
        if (shouldWeightX) {
            c.weightx = 0.5;
        }
        c.fill = GridBagConstraints.HORIZONTAL;
        c.gridx = 0;
        c.gridy = 0;
        pane.add(button, c);

        button = new JButton("Button 2");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 0.5;
        c.gridx = 1;
        c.gridy = 0;
        pane.add(button, c);

        button = new JButton("Button 3");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.weightx = 0.5;
        c.gridx = 2;
        c.gridy = 0;
        pane.add(button, c);

        button = new JButton("Long-Named Button 4");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.ipady = 40; //make this component tall
        c.weightx = 0.0;
        c.gridwidth = 3;
        c.gridx = 0;
        c.gridy = 1;
        pane.add(button, c);

        button = new JButton("5");
        c.fill = GridBagConstraints.HORIZONTAL;
        c.ipady = 0; //reset to default
        c.weighty = 1.0; //request any extra vertical space
        c.anchor = GridBagConstraints.PAGE_END; //bottom of space
        c.insets = new Insets(10,0,0,0); //top padding
        c.gridx = 1; //aligned with button 2
        c.gridwidth = 2; //2 columns wide
        c.gridy = 2; //third row
        pane.add(button, c);
    }

    private static void createAndShowGUI() {
```

```

//Membuat Frame
JFrame frame = new JFrame("GridBagLayoutDemo");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

//memanggil method addComponentsToPane
//untuk membuat panel dan menambahkan komponen
addComponentsToPane(frame.getContentPane());

//Menampilkan frame
frame.pack();
frame.setVisible(true);
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            createAndShowGUI();
        }
    });
}
}

```

Keterangan Method:

Method/Atribut	Fungsi
gridx, gridy	Menentukan baris dan kolom pada sudut kiri atas komponen. Koordinat paling kiri memiliki nilai gridx = 0; dan paling atas punya nilai gridy = 0; Gunakan GridBagConstraints.RELATIVE untuk penempatan sesuai pendeklarasian komponen terhadap layout
gridwidth, gridheight	Menentukan lebar atau panjang komponen terhadap layout. Nilai defaultnya adalah satu
ipadx, ipady	Menentukan besarnya komponen. Hampir sama dengan gridwidth/gridheight, hanya saja atribut ini tetap terhadap 1 nilai cell (padding).
insets	Menentukan space komponen dengan komponen atas, bawah, samping kiri dan samping kanan (spacing)
anchor	Memungkinkan peletakan komponen di koordinat tertentu layaknya border layout. FIRST_LINE_START PAGE_START FIRST_LINE_END LINE_START CENTER LINE_END LAST_LINE_START PAGE_END LAST_LINE_END Digunakan jika space content pane lebih besar dari keseluruhan komponen

9.5 LATIHAN

9.5.1 Soal

Buatlah sebuah tampilan program sebagai berikut menggunakan kombinasi layout flow (default layout dari JPanel), Grid, GridBag atau Border. Jumlah dan jenis layout (dari yang telah disebutkan) untuk digunakan tidak ditentukan.



The screenshot shows a Java Swing window titled "Form Tanggal Lahir". It is divided into two sections: "Input Biodata Lahir" and "Output Biodata Lahir".

Input Biodata Lahir:

- Tempat Lahir: Bandung (dropdown menu)
- Tanggal/Bulan/Tahun: 3 (dropdown menu), Pebruari (dropdown menu), 1990 (dropdown menu)

Output Biodata Lahir:

- Tempat Lahir: Bandung (text field)
- Tanggal/Bulan/Tahun: 3 (text field), Pebruari (text field), 1990 (text field)

Berikan Action untuk setiap pemilihan combo box. Setiap combo box terpilih, maka text field akan berisi nilai dari combo box.

9.5.2 Soal

Buatlah sebuah tampilan GUI sebagai berikut:



The image shows a Java Swing window titled "Biodata Mahasiswa". The window has a standard Mac OS-style title bar with red, yellow, and green window control buttons. The main content area contains the following elements:

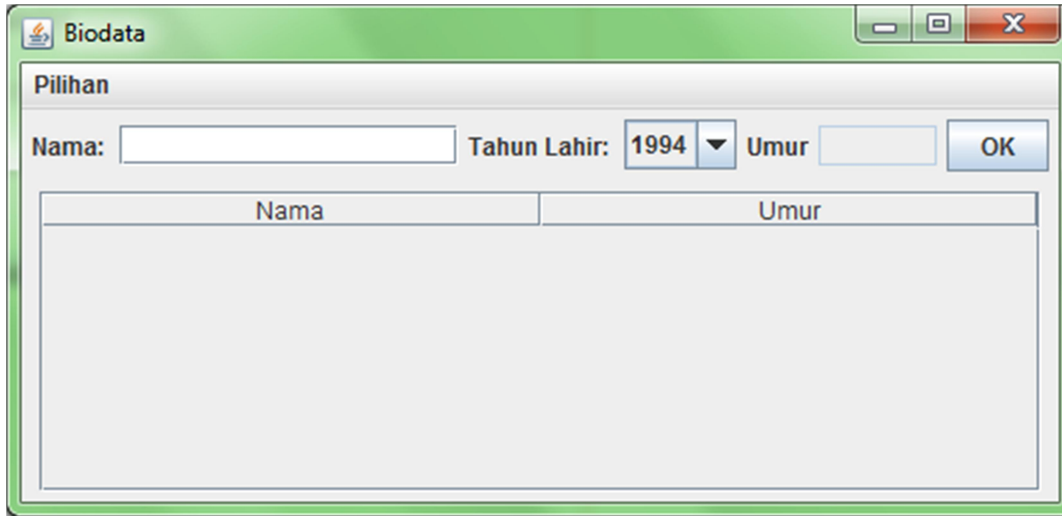
- A text label "NIM" followed by a text input field.
- A text label "Nama" followed by a text input field.
- A text label "Program Studi" followed by three radio buttons labeled "MI", "KA", and "TK".
- At the bottom of the window, there are three buttons: "Simpan", "Batal", and "Keluar".

Buatlah GUI dengan event handler dan petunjuk sebagai berikut:

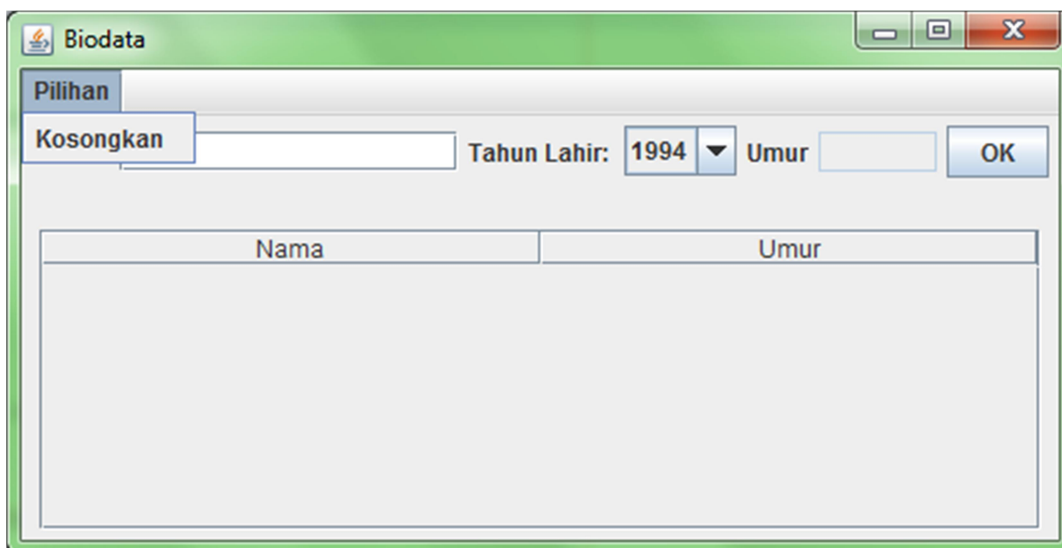
- Pada saat pengguna menekan tombol Simpan, maka nilai dari NIM, Nama, Pilihan Program Studi akan tersimpan/tampil di TextArea
- Pada saat pengguna menekan tombol Batal, maka nilai dari NIM, Nama, Pilihan Program Studi akan kosong
- Pada saat pengguna menekan tombol Keluar, maka form Biodata Mahasiswa akan tertutup

9.5.3 Soal

Buatlah sebuah form gui sebagai berikut tanpa menggunakan bantuan gui builder.



The screenshot shows a window titled "Biodata" with a green title bar. Inside, there is a tab labeled "Pilihan". Below the tab, there are three input fields: "Nama:" followed by an empty text box, "Tahun Lahir:" followed by a dropdown menu showing "1994", and "Umur" followed by an empty text box. To the right of these fields is an "OK" button. Below the input fields is a table with two columns: "Nama" and "Umur". The table is currently empty.



The screenshot shows the same "Biodata" window. The "Pilihan" tab is still selected. The "Nama:" input field now contains the text "Kosongkan". The "Tahun Lahir:" dropdown menu still shows "1994", and the "Umur" input field is empty. The "OK" button is still present. Below the input fields, the table with columns "Nama" and "Umur" remains empty.

Keterangan:

- Gui tidak terhubung dengan database
- Merupakan Class java yang meng-extends “JFrame” dan meng-implements “ActionListener”
- Frame memiliki 2 panel, panel atas (center) dan panel bawah (south)
 - Isi panel atas
 - 3 label
 - 2 textfield
 - 1 combobox
 - 1 button
 - Layout yang digunakan adalah layout default (flow layout)
 - Isi panel bawah
 - 1 Tabel